

Control and Observation in Distributed Environments

Warren Smith
Computer Sciences Corporation
NASA Ames Research Center

Motivation and Approach

- Large and distributed set of resources, services, applications
- There will be failures
- The grid must be managed
- Develop a general framework for observation and control
 - Observe and control a variety of resources and services
 - Support observation and control of user applications
- Extend the framework for specific tasks
 - Add components to observe new things
 - Add components to perform new actions
- Add new logic for management

Computational Grids

- Computational grids consist of distributed:
 - Resources
 - Services
 - Applications
 - Users
- Information Power Grid (IPG):
 - NASA's implementation of a persistent, secure, and robust grid
 - Collaborative design, analysis, and visualization environment
 - Coupled HPC resources
 - Real-time access to instruments



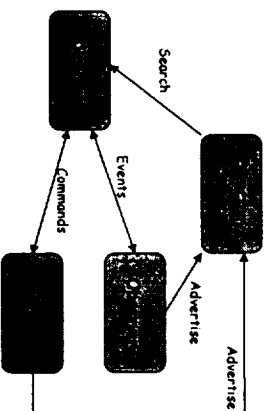
Why not use an existing system?

- Most existing monitors cannot be embedded in tools or applications
 - Examples: AIMS, Big Brother, and similar application/system monitors
 - Limited fault detection functionality
 - Examples: Heart Beat Monitor and Network Weather Service
 - System- or application-specific information but not both
 - Examples: SNMP based tools and MTRICH profiling
 - Lack of extensible data forwarding and gathering mechanisms
 - Example: Netlogger
 - Incompatibility with security and authentication requirements of IPG
- Deficiencies of existing monitors motivated us to develop our framework

Goals of our Infrastructure

- Develop a general framework for observation and control
 - ♦ Observe and control a variety of resources, services, and applications
 - ♦ Scalable
 - ♦ Secure
- Framework should be extensible for specific tasks
 - ♦ Add new components for observing and performing actions
 - ♦ Easily add new logic for management
 - ♦ Modular
- Compatible with emerging standards
 - ♦ Grid Forum Performance Working Group

High-Level Architecture



Observer



- Two low level components
 - ♦ Sensor
 - ♦ Event Service (Event Producer Interface)
 - Subscribe
 - Query
- Sensor manager
 - ♦ Manages sensors, subscriptions, and queries

Key

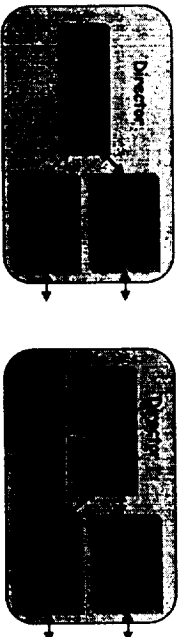
Actor



- Two low-level components
 - ♦ Actuator
 - ♦ Event Service (Actor Server Interface)
 - Actuator Manager
- Handles requests for actions

Key

Director



- One basic component - Event Service
- 2 approaches to higher-level components
 - ♦ User writes management logic
 - ♦ User writes management rules and uses an expert system

Directory Service

- Hold information about observers and controls
- Allows directors to find them
- Currently defining schema for the data

Implementation

- Written in C++
- Communicates using TCP, UDP, or SSL
- XML encoding of messages
- OpenSSL for authentication and security
 - ♦ Compatible with Globus Security Infrastructure
- PThreads
- CLIPS expert system
- Targeting IRIX, Solaris, Linux
- Port director code to Java for GUIs (future)

Uses of Infrastructure

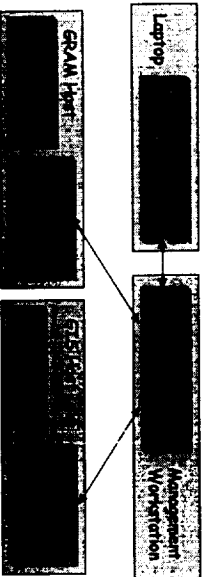
- Observe and control a Globus-based computational grid
- Information source for an alternative Grid Information Service
- Application performance analysis
- Application steering

Observation and Control of a Grid

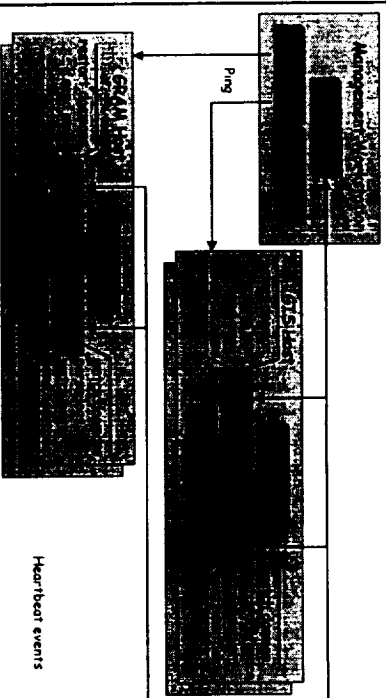
- As grids get larger, this becomes difficult
- Things to observe:
 - ♦ Grid Information Service servers
 - ♦ GRAM reporter daemons
 - ♦ Log files
 - ♦ Resource status and usage
- Things to control
 - ♦ Restarting/configuring GIS servers
 - ♦ Restarting GRAM daemons
 - ♦ Add/remove user mappings
 - ♦ Add/remove certificate authorities
- Provide a nice GUI to do all this

Grid Control Panel

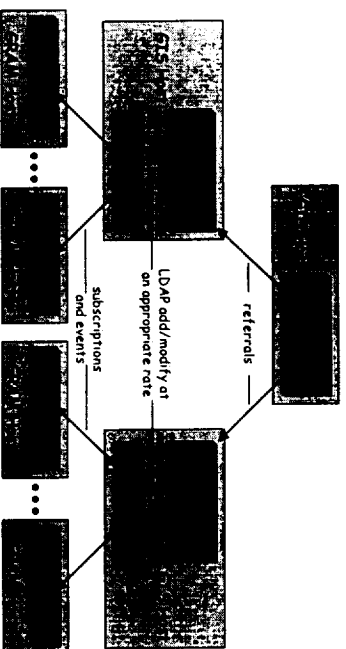
- Java GUI for managing a Grid
- Back end that is always running
- Front end GUI that the user starts
- Future work



Observation and Control of a Grid



An Alternative GIS



Standardization

- Performance Working Group of the Grid Forum
- Architecture
- Event representations
- Directory service schema
- Communication protocols
- This framework is compatible with the developing standards
 - <http://www.gridforum.org>
 - <http://www-didc.lbl.gov/GridPerf>

Status and Future Work

- Current Status:
 - ♦ Not yet released (amazing amount of paperwork)
 - ♦ C++ code designed, mostly implemented and tested, barely documented, but some rough edges
 - ♦ Preliminary GIS monitor and manager
- Our future plans include:
 - ♦ Improve security
 - ♦ Develop more sensors and actuators
 - ♦ More testing and documentation
 - ♦ Port appropriate modules to Java
 - ♦ Use the framework to build tools
 - ♦ Stay compatible with Grid Forum standards